UML DIAGRAMS

What is the UML?

The Unified Modeling Language is a visual language for specifying, constructing and documenting the artifacts of systems. The word visual in the definition is a key point -the UML is the de facto standard diagramming notation for drawing or presenting pictures. The standard is managed, and was created, by the Object Management Group. It was first added to the list of OMG adopted technologies in 1997.

UML is composed of 9 graphical diagrams:

1) **Class Diagram -** describes the structure of a system by showing the system's classes, their attributes, and the relationships among the classes.

2) **Use – Case Diagram -** describes the functionality provided by a system in terms of actors, their goals represented as use cases, and any dependencies among those use cases.
3) **Behavior Diagram**

   a. **Interaction Diagram**
      i. **Sequence Diagram -** shows how objects communicate with each other in terms of a sequence of messages. Also indicates the lifespan of objects relative to those messages.
      ii. **Communication diagram:-** how the interactions between objects or parts in terms of sequenced messages.

   b. **State Chart Diagram -** describes the states and state transitions of the system.

   c. **Activity Diagram -** describes the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

4) **Implementation Diagram**
   a. **Component Diagram -** describes how a software system is split up into components and shows the dependencies among these components.

b. **Deployment Diagram -** describes the hardware used in system implementations and the execution environments and artifacts deployed on the hardware.

Three Ways to Apply UML

- **UML as sketch :** Informal and incomplete diagrams created to explore difficult parts of the problem or solution space, exploiting the power of visual languages.
- **UML as blueprint :** Relatively detailed design diagrams used either for

  1) Reverse Engineering : UML tool reads the source or binaries and generates UML package, class, and sequence diagrams to visualize and better understanding of existing code in UML diagrams .

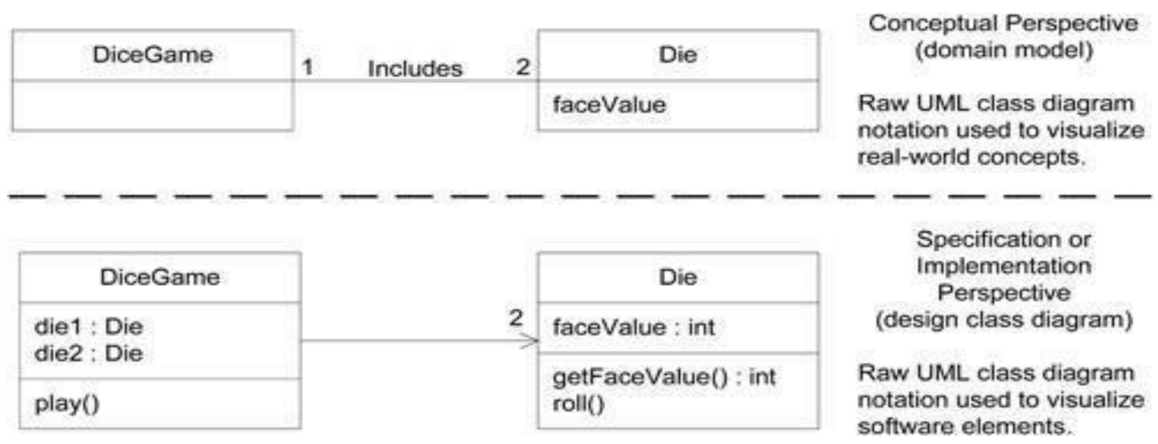
  2) Forward Engineering : code generation . Before programming, some detailed diagrams can provide guidance for code generation (e.g., in Java), either manually or automatically with a tool. It's common that the diagrams are used for some code, and other code is filled in by a developer while coding

- **UML as programming language :** Complete executable specification of a software system in UML. Executable code will be automatically generated

Three Perspectives to Apply UML

The same UML class diagram notation can be used to draw pictures of concepts in the real world or software classes in Java.

1. **Conceptual perspective -** the diagrams are interpreted as describing things in a situation of the real world or domain of interest.
2. **Specification (software) perspective -** the diagrams (using the same notation as in the conceptual perspective) describe software abstractions or components with specifications and interfaces, but no commitment to a particular implementation (for example, not specifically a class in C# or Java).
3. **Implementation (software) perspective -** the diagrams describe software implementations in a particular technology (such as Java).


DiceGame
1
Includes
2
Die
faceValue
Conceptual Perspective (domain model)
Raw UML class diagram notation used to visualize real-world concepts.
DiceGame
die1 : Die
die2 : Die
play()
2
Die
faceValue : int
getFaceValue() : int
roll()
Specification or Implementation Perspective (design class diagram)
Raw UML class diagram notation used to visualize software elements.

The Meaning of "Class" in Different Perspectives

- **Conceptual class** real-world concept or thing. A conceptual or essential perspective.

  The UP Domain Model contains conceptual classes.
- **Software class** a class representing a specification or implementation perspective of a software component, regardless of the process or method.
- **Implementation class** a class implemented in a specific OO language such as Java.